HONEYWELL SOFTWARE SUMMARY

SERIES 200

NEW DIMENSIONS OF PROVEN COMPUTER PERFORMANCE WITH EXCEPTIONAL ABILITY TO MATCH THE EXACT DIMENSIONS OF YOUR BUSINESS

INTRODUCTION

The hardware of Honeywell's Series 200 Electronic Data Processing System reflects the most advanced know-how in today's computer industry. To consummate the effectiveness of this system, Honeywell has developed a comprehensive array of programming and operating aids, or "software," that will ensure each user's realization of the hardware's great potential.

COMPREHENSIVENESS

Series 200 software is comprehensive. All the functions required for normal data processing are included in the library of programming and operating aids.

OBJECT PROGRAM PREPARATION — Conversion of programs from one of several easy-to-use source languages, or from certain competitive system languages, to machine code.

GENERALIZED DATA MANIPULATION — The performance of those data processing chores common to most users, such as sorting, input/output operations, and report generation.

MAINTENANCE AND OPERATION OF PROGRAMS — Updating of object program master files, selection of programs to be run, and control of checkout and production operation.

GENERAL UTILITY FUNCTIONS — Auxiliary operations such as tape searching, tape file copying, media conversion, and dynamic memory and tape dumps.

PROGRAM COMPATIBILITY

A distinctive feature of Series 200 programming aids is that they and the object programs which they produce are operationally compatible with one another. This property enables the operating system to draw all elements into an integrated whole. Object programs produced by a variety of program preparation aids, as well as programs from the software library itself, may all be intermixed on run tapes and processed by the PLUS Operating System. The advantages offered by this feature are highlighted in a succeeding discussion of the PLUS System.

Complete program compatibility is a built-in feature of Series 200. A single machine language is used with all of the Models 120/200/1200/2200/4200, allowing the user to run on the Model 4200 any program written for the smaller machines. Thus, software and software-produced object programs which run on even the smallest possible processor can also run on the larger configurations, and usually with a considerable gain in performance due to the faster cycle times and increased input/output simultaneity.

FLEXIBILITY

The outstanding modularity and resultant flexibility of Series 200 hardware have their parallels in the programming and operating aids furnished with the system. Most types of programs in the software library are offered in several versions to run in systems configurations of different sizes and compositions. In particular, it is important to note that software versions written for large systems are designed to take advantage of the increased internal and input/output processing capacities of these systems. In addition, the majority of software programs utilizing punched cards are also implemented for punched paper tape. Furthermore, a comprehensive array of random access drum software is provided.

PROGRAM PREPARATION AIDS

Honeywell supplies the Series 200 user with an assembly system, a program conversion system, and compilers. Specifically, these comprise the Easycoder Assembly System, the Liberator conversion programs Bridge and Easytran, a COBOL (Common Business-Oriented Language) compiler, and Fortran, a scientific-language compiler.

Of considerable importance to users is the ability of the assembler and compilers to produce programs tagged as belonging to a particular "job." To make use of this facility, the programmer need only specify a particular parameter (e.g., A) in the director cards for all of the programs constituting a job. Then at object time, programs may be called and executed individually; or a complete job, consisting of several programs, may be called and automatically executed in its entirety. The programmer may assign several parameters to a single program so that it becomes part of a number of different jobs.

EASYCODER ASSEMBLY SYSTEM

An assembly system comprises two elements: a symbolic language, and an assembly program which translates source programs written in the symbolic language into machine language. The Series 200 Assembly System, called Easycoder, is furnished in versions to meet the needs of all system sizes, from the smallest to the largest configuration.

EASYCODER FOR SMALL SYSTEMS

The versions of Easycoder for small, card-oriented and paper tape-oriented systems are designed for configurations which include from 2,048 to 12,288 characters of core memory and a card (or paper tape) reader and punch. Magnetic tape drives, if available, can be used to advantage in these configurations. However, programs assembled by these versions can be run on a system of any size. Specifically associated with these versions of Easycoder are:

CARD LOADER, PAPER TAPE LOADER, and MEMORY DUMP — Routines generated automatically according to a programmer's specification and included in an object program to perform, respectively, program loading operations and a printout of memory contents upon request.

CONDENSE — A routine which accepts an object program deck produced by Easycoder Assembly, each card of which contains a single instruction, and produces a condensed deck containing several instructions per card. A typical deck is compressed to about one-fifth its original size. (Requires a 4,096-character memory, a card reader or magnetic tape unit, and a card punch or magnetic tape unit.)

Even in a minimum equipment configuration, Easycoder assembles programs at an average rate of 110 program statements per minute on the Model 200. This speed can be increased to 450 statements per minute in systems which include 4,096 characters of memory and two tape units to receive output.

EASYCODER FOR LARGER SYSTEMS

The Easycoder Assembly System takes advantage of larger equipment configurations to provide additional flexibility and speed. In addition to advanced language functions (described below), Easycoder for larger systems also provides:

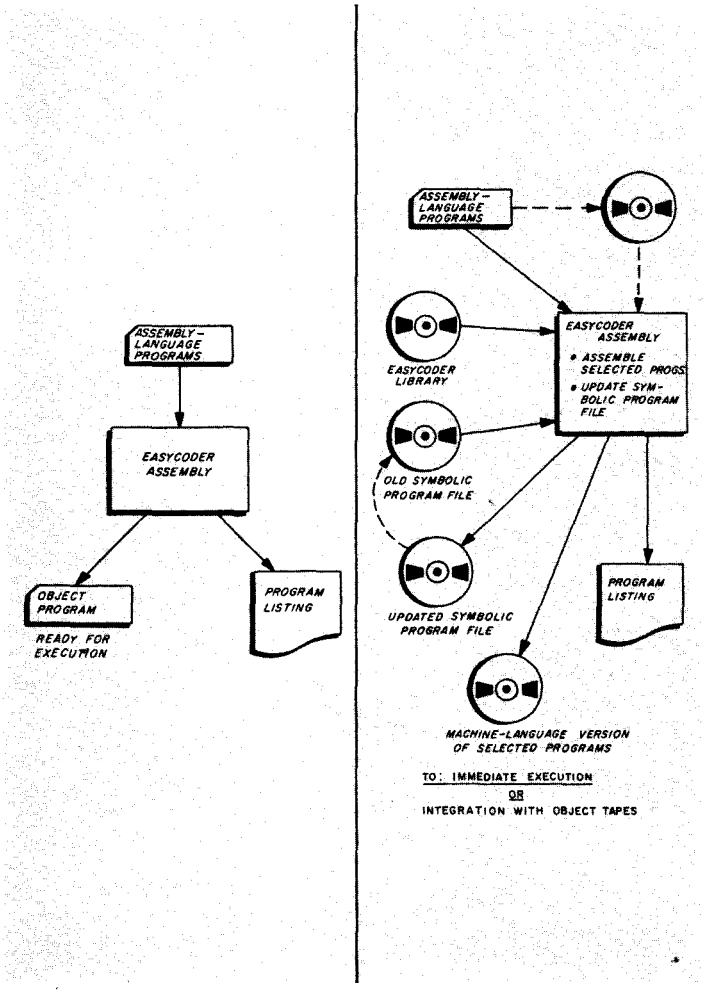
MAINTENANCE OF SYMBOLIC PROGRAM FILE — A file of Easycoder source programs is input to each assembly run and is updated as specified by the programmer.

SELECTIVE ASSEMBLY — Run tapes contain specific programs selected by the programmer from both an input deck of new programs and a tape file of previously processed symbolic programs.

LIBRARY FACILITIES — A basic library of general-purpose software is furnished by Honeywell to perform common jobs; to this the user can add his own often used programs and routines. Programs in the library can be conveniently assembled into an object program by the use of macro instructions.

HIGH PERFORMANCE — The use of highly efficient processing techniques, in conjunction with the hardware power of Series 200, enables Easycoder for larger systems to achieve very high levels of performance (up to 1000 statements per minute on the Model 200).

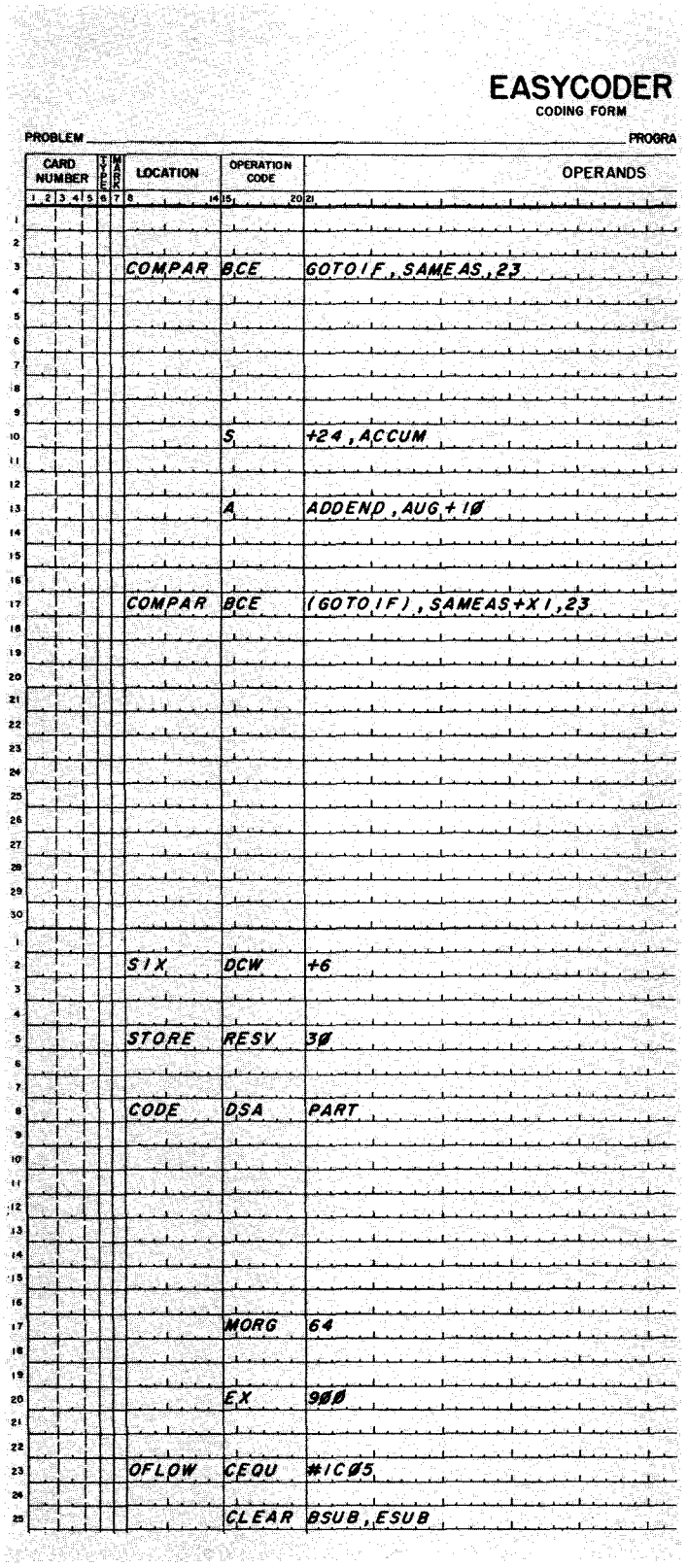
ENVIRONMENTAL FLEXIBILITY — Easycoder for larger systems can be utilized in both tape- and card-oriented installations.



EASYCODER ASSEMBLY LANGUAGE

Honeywell's Easycoder Assembly System includes an assembly language which combines ease of use with power and flexibility. The Easycoder assembly language incorporates easily remembered mnemonic operation codes. Memory location addresses may be denoted either by absolute decimal numbers or by symbolic tags; a symbolic address may be expressed relative to a tag defined elsewhere in the same program, e.g., as TAG+4. Both indexed and indirect addressing may be specified in Easycoder Language.

The value of an operand may be specified directly by means of a literal. Also, programming is greatly simplified through the use of macro instructions which cause the generation of appropriate sequences of machine instructions or the insertion of a library routine into an object program.



EASYCODER
CODING FORM

PROBLEM _____ PROGRA

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS
1 2 3 4 5 6 7 8			14 15	20 21	
			COMPAR	BCE	GOTOIF,SAMEAS,23
				S	+24,ACCUM
				A	ADDEND,AUG+10
			COMPAR	BCE	(GOTOIF),SAMEAS+X1,23
			SIX	DCW	+6
			STORE	RESV	30
			CODE	DSA	PART
				MORG	64
				EX	900
			OFLOW	CEQU	#IC05
				CLEAR	BSUB,ESUB

MACHINE INSTRUCTION STATEMENTS (direct counterparts of Series 200 machine instructions) — Typical statement consists of tag, mnemonic operation code, one or more operand addresses (or literal operands), and one or more variant characters.

BCE (Branch if Character Equal) statement. Referenced elsewhere by the symbolic tag COMPAR; assembly produces machine instruction to cause a branch to location GOTOIF if location SAMEAS contains the octal value 23.

Subtract statement using literal operand; 24 to be subtracted from contents of ACCUM.

Relative addressing; statement adds contents of ADDEND to field 10 locations beyond AUG.

Above BCE instruction modified to illustrate indexed and indirect addressing. Address of data tagged SAMEAS is added to contents of index register 1 (X1) to form address for comparison. Data at address formed is compared to variant (23), and if equal, program branches to location whose address is stored at GOTOIF (indirect addressing).

DATA FORMATTING STATEMENTS — These statements instruct the Assembly Program to set up constants and reserved memory areas and to punctuate memory to indicate field boundaries.

Store the decimal value +6 as a constant in a location to be addressed by the tag SIX.

Reserve an area of 30 locations to be referenced by the tag STORE.

Store the absolute address assigned to the tag PART in a field which can be referenced by the tag CODE.

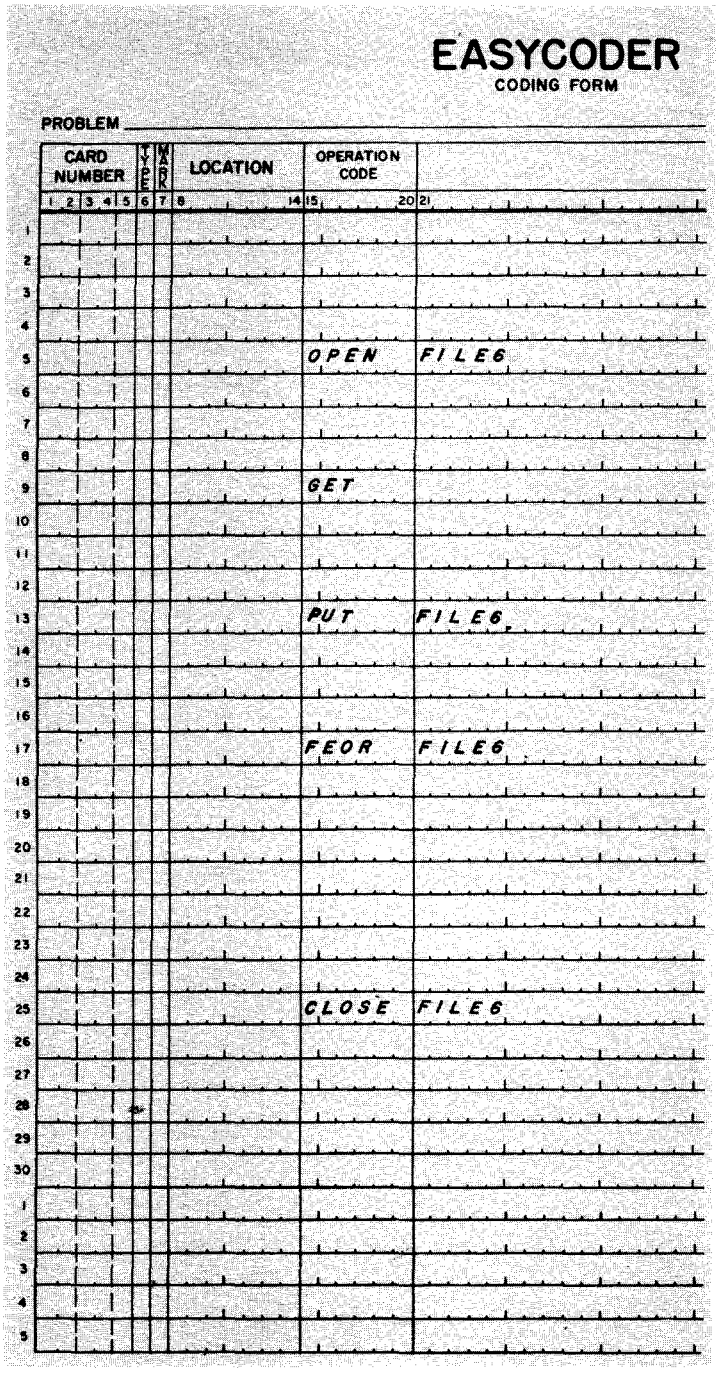
ASSEMBLY CONTROL STATEMENTS — These statements instruct the Assembly Program in the performance of a wide variety of functions related to creating an object program.

Memory allocation for subsequent statements is to begin with the next location whose address is a multiple of 64.

Provide for a temporary interruption of object program loading to branch to location 900.

Equate the tag OFLOW to the octal value 5.

Provide for clearing to zeros the memory area from BSUB to ESUB before loading the object program.



TIPTOP MACRO INSTRUCTIONS AND RESULTING OPERATIONS

Open Tape File — Prepares the specified file to be read or written. Reads and checks input header labels and writes new labels for output file.

Get Item — Makes the next item from the specified file available to the program for processing; a depleted input buffer is filled automatically by reading a record.

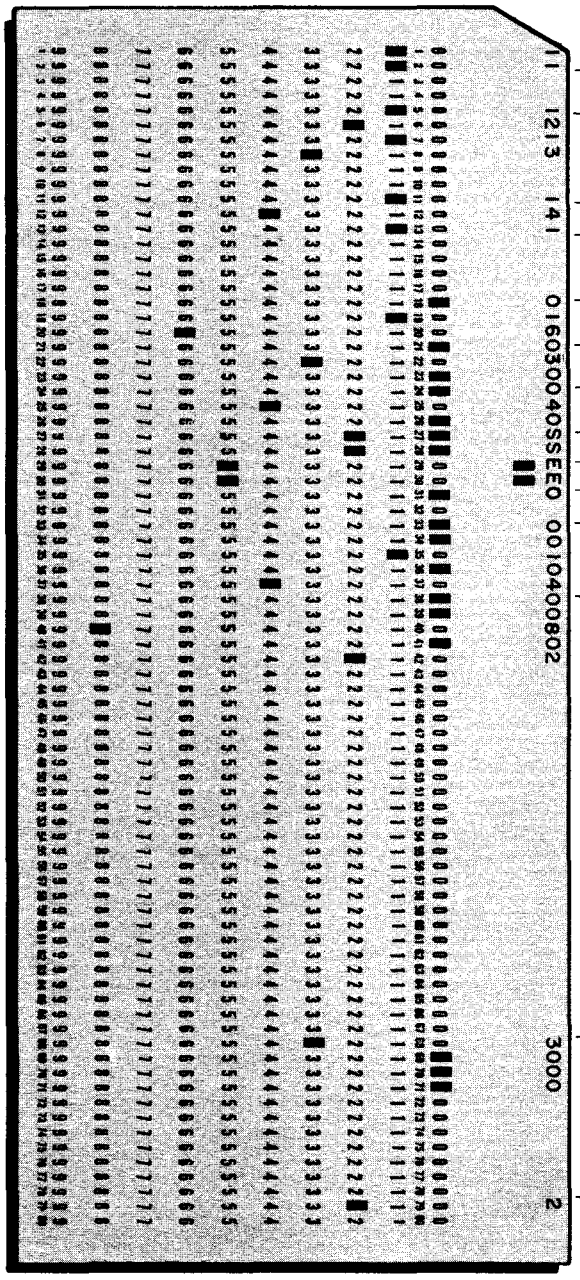
Put Item — Moves a processed item into the next output buffer available; items from full buffer are automatically transferred to tape.

Force End-of-Reel Condition — Forces premature occurrence of the operations normally performed automatically when the physical end of reel is reached. These operations include: rewinding tape, incrementing the reel sequence number, writing tape labels (output file only), and swapping tapes if this action is specified in the TIPTOP input parameters.

Close File — If input file, deactivates the file, rewinds the tape if specified in TIPTOP input parameters, and exits to main program. If output file, pads record in output buffer if short and transfers it to tape, writes labels, exits to programmer-specified routines, rewinds tape, and exits to main program.

TAPE SORT AND COLLATE PROGRAMS

These are generalized programs which adapt themselves, as directed by programmer-specified parameters, to operate in a particular hardware configuration and to sort and collate data in a particular format. All of the sort programs take advantage of the industry-acclaimed Polyphase sorting technique developed by Honeywell. Tailored for use in small systems is a sort program which requires only three tape units and receives its specialization parameters by card or paper tape. This program sorts fixed-length records on up to seven keys and provides facilities for own-coding.



SORT PARAMETER CARD

11 — Address of input tape.
12, 13 — Addresses of work tapes.
14 — Address of merge work tape.
1 — Total number of input reels.
016 — Number of characters per item.
030 — Number of items per input record.
040 — Number of items per output record.
S, S — Std. beginning and end-of-file labels.
EE — Even-parity input and output.
0 — Padding character for short end record.
00104 — First key starts in 1st position and is 4 characters long.
00802 — Second key starts in 8th position and is 2 characters long.
3000 — Address to which presort should branch prior to processing each item.
2 — Halt on uncorrectable read error.

More advanced sort programs are furnished for use in larger systems. These programs provide the added advantages of read-backward Polyphase sorting and the ability to handle variable-length records. A sort program can be automatically linked to a series of related operations by coding the preceding program to establish the sort parameter values before it terminates. These programs are also self-adapting to memories larger than the minimum.

For use in conjunction with each of the sort programs, when needed, a collate program is available. The collate programs accept two or more sorted files and combine them to produce a single composite file in proper sequence.

SORT PROGRAM PERFORMANCE

The sort times in the accompanying table exemplify the power of Series 200 sorts. The information given is from an actual situation in which a file, submitted by a customer for testing, was sorted by one of the ad-

vanced sort programs mentioned above. The file, consisting of 8850 randomly distributed, 80-character, unblocked items each having a single 5-character key field, was sorted by a Model 200 with 66,700-character-per-second tape units.

Memory Capacity in Characters	Number of Tape Units	Actual Run Times in Minutes
8,192	3	5.8
8,192	4	4.9
12,288	3	5.0
12,288	4	4.0

SCIENTIFIC SUBROUTINES

Series 200 scientific users have available to them an extensive library of scientifically oriented subroutines which complement the capabilities of the Fortran compilers. This library includes the usual basic Fortran routines, such as square root, exponential, trigonometric, and logarithmic functions, as well as matrix, statistical, and other more comprehensive routines. All the subroutines in this library can be used with or without the scientific hardware option.

TABULATING EQUIPMENT SIMULATION—TABSIM

A tabulating equipment simulator, TABSIM prepares printed reports from input consisting of a deck of punched cards (or a tape file of card images). The input deck contains control cards and data on detail cards. The control cards contain instructions for the processing of the detail cards. In general terms, the output report represents the data on the detail cards, edited and processed arithmetically.

REPORT GENERATION

Honeywell furnishes a program for automatic creation of reports according to user specifications. To use the report generator, the programmer merely prepares a set of parameters defining control fields and report lines. These parameters are used as input to the report generator, which produces a symbolic program. The assembled version of this program accepts raw data from cards or tape, edits it, and generates the desired reports.

COMMUNICATION SOFTWARE

The high internal speeds of the Series 200, along with the hardware program interrupt feature, make it well suited to communication and other real time applications. Honeywell is furnishing a complete set of software to monitor communication activities in the system. In particular, the following types of routines are being provided:

INTERRUPT — Entered automatically upon the occurrence of a program interrupt, this routine directs the transfer of data between a communication control and the central processor and then returns control to the main program.

MESSAGE QUEUING — Controls the order in which messages are stored, processed, and transmitted.

ERROR CONTROL — Corrects, wherever possible, errors in messages received from other communication stations.

SOFTWARE FOR RANDOM ACCESS DEVICES

Honeywell offers a comprehensive array of programming and operating aids for the Series 200 drum user, including a Drum Loader/Monitor, a program for updating program files on a drum, a special drum sort, input/output routines, and utility routines.

RANDOM ACCESS SORT

A separate program is furnished to sort data stored on magnetic drums. This program strips off the item keys of data stored on a drum, sorts the keys, and then stores on the drum a table containing the keys and the addresses of the corresponding file items. Items may be brought in from the drum in the order of the sorted keys by using the Easycoder macro instruction FETCH.

RANDOM ACCESS I/O PACKAGE — DIPDOP

Direct, serial, and random processing of drum files are provided by this control package. Easycoder macro instructions are available to direct the performance of the following drum input/output functions:

DIRECT-ADDRESS PROCESSING — Reading or writing of data from a sector whose address is given.

SERIAL PROCESSING — Reading or writing of the item following the one currently being processed.

RANDOM PROCESSING — Transfer of an item between core memory and a drum location whose address is determined by mathematical transformation of the item's key.

The drum input/output package processes either fixed- or variable-length items and blocks and unblocks items within records. To assist the user further, it also affords facilities for detection and automatic correction of errors.

RANDOM ACCESS UTILITY ROUTINES

Honeywell has designed a "package" of generalized utility routines for use at drum installations. The jobs performed by these routines include the following:

- *Examining the contents of a drum file.*
- *Transferring a file between a drum and punched cards or magnetic tape (the transfer may be in either direction).*
- *Making corrections to a file stored on a drum.*

The separate routines which perform these functions may be assembled with a control routine to form an independent system called DIAL (Drum Interrogation, Alteration, and Loading); or individual routines may be assembled directly into an object program. In particular, Easycoder users may obtain specified DIAL functions by the use of macro instructions such as LOCATE, UNLOAD, RESTORE, EDIT, CORRECT, COMPARE, and CLEAR.

OPERATING SYSTEM — PLUS

The PLUS Operating System is the central integrating element for all programs running in medium- and large-scale Series 200 installations. It enables object programs generated by the program-preparation aids to be combined with one another, and with Honeywell-supplied routines, to produce run tapes, drum files, or binary card decks containing all user-required functions. The key to this flexibility is the previously noted compatibility of object programs produced by the Series 200 software.

The PLUS System comprises a series of control and utility routines. These programs and routines can be combined to meet the specific requirements of each user while providing for the systematic exploitation of the equipment capabilities.

CONTROL PROGRAMS

Control programs form the core of the PLUS Operating System. They provide for the automatic processing of sequential programs during checkout or production runs. Under directions from the operator or from a running program, the control programs handle such functions as loading, segmentation control, and library search. In addition, they also control space and time sharing among several jobs running concurrently. They handle all communication between the operator and the running programs and provide for the effective coordination of all system operations from a single location.

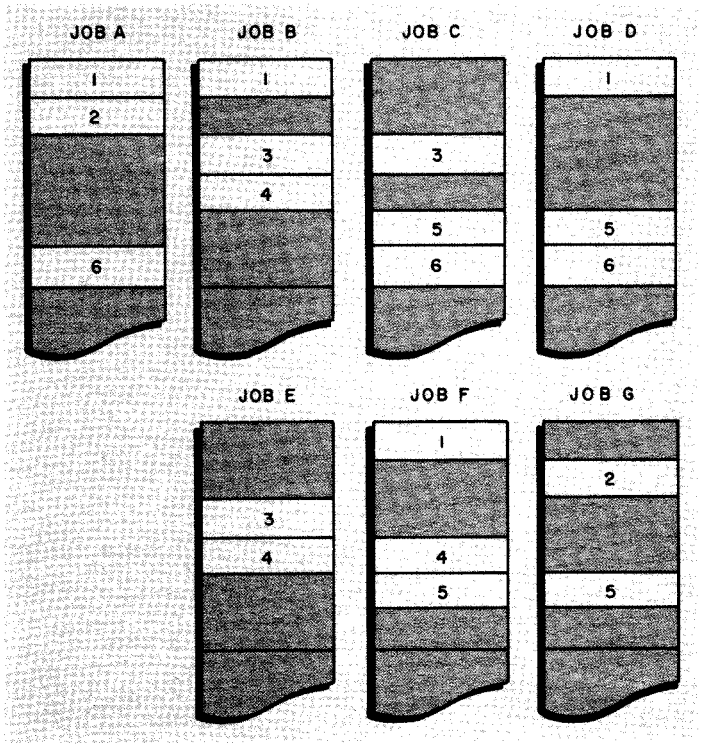
JOB-ORIENTED OPERATION

In the loading operation for a medium-scale system, for example, programs may be called singly or by "job." That is, an instruction may be given to execute a program with a given name, or to perform an entire job consisting of several programs. By way of illustration, assume that a run tape is being used which contains programs having the following job assignment parameters:

ORDINAL POSITION OF PROGRAM ON TAPE	JOB ASSIGNMENT PARAMETERS
1	A B D F
2	A G
3	B C E
4	B E F
5	C D F G
6	A C D

If the operator calls job C, the program tape is searched until the first program belonging to job C, which is program 3, is found. The program is loaded and executed, followed automatically by programs 5 and 6, which constitute the remainder of job C.

In effect, the specification by the operator of a job to be done makes certain programs on tape (viz., those constituting the specified job) "visible" to the Loader. Thus, the tape in the example above, although containing only six programs, contains seven jobs, as indicated in the accompanying illustration.



LIBERATOR CONVERSION PROGRAMS

The Liberator concept is a unique design criterion of the Honeywell Series 200. It allows the users of several competitive systems to take advantage of the superior throughput and cost/performance characteristics of the Honeywell equipment without incurring the prohibitive costs of reprogramming. Stated simply, Liberator is a hardware/software design concept which makes the basic instruction repertoire of the Series 200 equivalent to the instruction repertoires of several other data processing systems, viz., the IBM 1400 series. Automatic conversion programs are provided to specialize the program compatibility of the Series 200 to the requirements of programs written for competitive systems.

BRIDGE — CONVERSION OF MACHINE-LANGUAGE PROGRAMS

Bridge accepts a 1400-series, machine-language program as a source program, analyzes and transforms it into a Honeywell machine-language object program, and produces a composite listing of the competitive and Honeywell programs with diagnostics. Run times for converted programs are frequently several times faster than those obtainable on competitive systems, due primarily to two factors: the superior power of Series 200 hardware; and provisions made by Bridge to take advantage of the Honeywell capability for simultaneous input/output and central processor operations.

A portion of a composite listing produced during Bridge conversion of a 1401 source program is shown below.

SAMPLE OF COMPOSITE LISTING PRODUCED BY 1401 BRIDGE

```
PROGRAM  TEST36

            DEC   1401 MACHINE         OCTAL  MNMC
SEQ    CT   LOCN  INSTRUCTION          LOCN   CODE  HONEYWELL OBJECT INSTRUCTION     FLAG

00431  07  03181  L H32 274            006155  LCA     15  007370  000422
00432  07  03188  L H32 249            006164  LCA     15  007370  000371
00433  07  03195  E Q66 246            006173  MCE     74  005462  000366
00434  07  03202  A Q66 Q56            006202  A       36  005462  005450
00435  07  03209  E Q56 271            006211  MCE     74  005450  000417
00436  08  03216  B C43 244            006220  BCE     55  006417  000364 15
00437  08  03224  B C58 268            006230  BCE     55  006436  000414 15
00438  01  03232  2                    006240  CSM     61
00439  01  03233  /                    006241  CSM     13
00440  01  03234  /                    006242  CSM     13
00441  07  03235  M H13 Q66            006243  MCW     14  007345  005462
00442  07  03242  M H13 Q56            006252  MCW     14  007345  005450
00443  04  03249  B Q87                006261  B       65  005507
00444  01  03253  N                    006265  NOP     40
00445  04  03254  B D20                006266  B       65  006534
00446  04  03258  , B54                006272  SW      22  006266
00447  02  03262  F L                  006276  CSM     35 43
00448  07  03264  M H37 219            006300  MCW     14  007375  000333
00449  07  03271  L H29 271            006307  LCA     15  007365  000417
00450  07  03278  L H29 246            006316  LCA     15  007365  000366
00451  07  03285  E Q86 246            006325  MCE     74  005506  000366
00452  07  03292  A Q86 Q76            006334  A       36  005506  005474
00453  07  03299  E Q76 271            006343  MCE     74  005474  000417
00454  08  03306  B C43 244            006352  BCE     55  006417  000364 15
00455  08  03314  B C58 268            006362  BCE     55  006436  000414 15
00456  01  03322  2                    006372  CSM     61
00457  01  03323  /                    006373  CSM     13
00458  01  03324  /                    006374  CSM     13
00459  07  03325  M H13 Q86            006375  MCW     14  007345  005506
00460  07  03332  M H13 Q76            006404  MCW     14  007345  005474          OTP
00461  04  03339  B D20                006413  B       65  006534
00462  04  03343  H C57                006417  CSM     00  006435
00463  07  03347  M I86 246            006423  MCW     14  007622  000366
00464  04  03354  B 000                006432  B       65  000000            2
00465  04  03358  H C72                006436  CSM     00  006454
00466  07  03362  M I86 271            006442  MCW     14  007622  000417          OTP
00467  04  03369  B 000                006451  B       65  000000            2
00468  07  03373  ) +89 A73            006455  CW      23  006021  006145
00469  07  03380  ) B54 F21            006464  CW      23  006266  007045
00470  04  03387  B R96                006473  B       65  005664
00471  04  03391  H D19                006477  CSM     00  006533
00472  07  03395  H 094 000            006503  CSM     00  000020  000000     2
00473  08  03402  M UZ U63 W           006512  CSM     41  746402  002643 66
00474  01  03410  N                    006522  NOP     40
```

The entries on the left side of the listing describe the original 1401 instructions. The first of these represents a 1401 Load instruction which would normally be loaded into location 3181 and would initiate a data transfer between locations H32 and 274. The entries on the right side of the listing describe the Series 200 instructions created by Bridge which correspond to the left-side entries. The expression LCA, under the heading MNMC (Mnemonic) CODE, is the Easycoder symbolic code for the Honeywell instruction Load Characters to A-Field Word Mark, which corresponds to the 1401 Load instruction. The value 15 immediately following LCA is the Honeywell machine-language equivalent of the Load operation code. Entries under the heading Flag indicate conditions in the converted program which may be of interest to the user.

EASYTRAN — SOURCE-PROGRAM CONVERSION

Implemented for both 1400-series systems and the Honeywell Series 200, Easytran converts SPS or Autocoder programs into Easycoder programs. Since Easytran can operate on competitive systems, it enables many prospective users to convert their programs prior to delivery of their Honeywell systems. Thereafter, all that is required to produce operating programs is that the converted programs be processed by the Easycoder Assembly System.

Easytran provides an output listing which contains both the source coding and the resulting Easycoder symbolic coding. The upper part of the illustration shows the initial portion of a listing produced during the conversion of a 1401 program in a typical Easytran run. Detailed diagnostic information is provided by Easytran on a separate page of the listing. For example, the diagnostic information associated with the Easycoder program entry having the SEQ number 2275 is included in the lower part of the accompanying illustration.

To assist the user further, Easytran provides a cross-reference listing of all symbolic tags used in the input program, indicating all references made to the tags. The user checks for the presence of flags and performs any manual modification which may be required. He then uses the converted card deck as input to the Easycoder Assembly System, which produces a Honeywell object program.

REPRESENTATIVE EASYTRAN LISTING

```
              EASYTRAN LISTING          THIS IS A TYPICAL EASYTRAN RUN

                                 EASYCODER                                              AUTOCODER

REF SEQ   T M  LOCN    OP    OPERANDS                          S  LABEL   OP    OPERANDS

    2252                CW    COMP+4,STSCOR+1                               CW    COMP6004,STSCOR6001
    2253                CW    ZEROSC+4,STSCOR+4                             CW    ZEROSC6004,STSCOR6004
    2254                CW    COMP+1                                        CW    COMP6001
    2255                B     COMP                                          B     COMP
  M 2256       PRINT    MCW   0,H,PRNT+3                          PRINT     MLC   0000,0203
    2257                CSM   0                                             W
    2258                DC    *4C00010006
    2259                C     COMP+3,LSTTBL                                 C     COMP6003,LSTTBL
    2260                B     READCD,42                                     BE    READCD
    2261                SW    PRINT+1                                       SW    PRINT6001
    2262                A     N3,HOLD                                       A     N3,HOLD
W   2263                BA    N2,PRINT+3                                    A     N2,PRINT6003
    2264                CW    PRINT+1                                       CW    PRINT6001
    2265                B     RTN                                           B     RTN
    2266       XRTN     CSM   H,PRNT+85,H,PRNT,03                 XRTN      CS    0285
    2267                MCW   LOC224,XSTSCR+6                               MLC   LOC224,XSTSCR6006
    2268                MCW   LOC227,STP+6                                  MLC   LOC227,STP6006
    2269                MCW   SCOREL,XCOMP+6                                MLC   SCOREL,XCOMP6006
    2270                MCW   LOC23,XSTSCR+3                                MLC   LOC23,XSTSCR6003
    2271       XRTN1    MCW   HOLD,XCOMP+3                        XRTN1     MLC   HOLD,XCOMP6003
    2272                MCW   LMARG,STP+3                                   MLC   LMARG,STP6003
    2273                A     N42,HOLD                                      A     N42,HOLD
  M 2274       XCOMP    C     0,0                                 XCOMP     C     0000,0000
    2275                B     XSTSCR,44                                     BH    XSTSCR
W   2276                C     XCOMP+3,HOLD                                  C     XCOMP6003,HOLD
    2277                B     STP+7,42                                      BE    STP6007
    2278                SW    XCOMP+1,STP+1                                 SW    XCOMP6001,STP6001
W   2279                BA    N3,XCCMP+3                                    A     N3,XCOMP6003
W   2280                BA    N2,STP+3                                      A     N2,STP6003
    2281                CW    XCOMP+1,STP+1                                 CW    XCOMP6001,STP6001
    2282                B     XCOMP                                         B     XCOMP
  M 2283       XSTSCR   MCW   0,0                                 XSTSCR    MLC   0000,0000
  M 2284       STP      MCW   0,0                                 STP       MLC   0000,0000
    2285                SW    STP+4,XCOMP+4                                 SW    STP6004,XCOMP6004
    2286                SW    XSTSCR+1,XSTSCR+4                             SW    XSTSCR6001,XSTSCR6004
W   2287                BA    N6,STP+6                                      A     N6,STP6006
W   2288                BA    N3,XCCMP+6                                    A     N3,XCOMP6006
W   2289                BA    N3,XSTSCR+3                                   A     N3,XSTSCR6003
W   2290                BA    N6,XSTSCR+6                                   A     N6,XSTSCR6006
    2291                C     XSTSCR+3,LOC50                                C     XSTSCR6003,LOC50
    2292                CW    STP+4,XCOMP+4                                 CW    STP6004,XCOMP6004
    2293                CW    XSTSCR+1,XSTSCR+4                             CW    XSTSCR6001,XSTSCR6004
    2294                B     PRINTX,41                                     BL    PRINTX
    2295                B     XRTN1                                         B     XRTN1
    2296       PRINTX   MCW   H,READ+20,H,PRNT+20                 PRINTX    MLC   0020,0220
    2297                CSM   0                                             CC    S
    2298                DC    *4C02000011
    2299                CSM   0                                             W     READCD
    2300                DC    *4C00010006
    2301                B     READCD
    2302       LOC50    DSA   H,READ+50                           LOC50     DSA   0050


        EASYTRAN DIAGNOSTIC WORKSHEET


          BRANCH HIGH BRANCH LOW
          THE FOLLOWING ENTRIES REFERENCE BRANCH HIGH OR BRANCH LOW
        INSTRUCTIONS. POSSIBLE OPERATING PROBLEMS, DUE TO DIFFERENCES
        BETWEEN THE 1401 AND THE HONEYWELL CHARACTER CODES, SHOULD BE CHECKED.

               006 2238
               007 2275
               007 2294
```

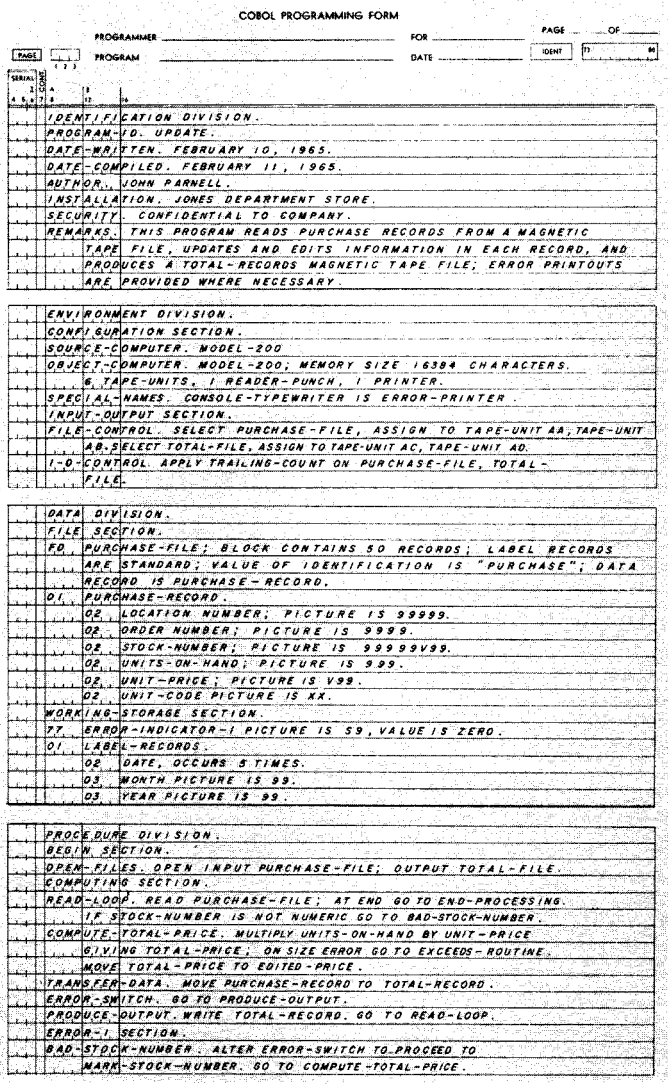
SERIES 200 COBOL

COBOL consists of a language which is a standardized, business-oriented subset of English and a processing system called a compiler. The programmer describes a solution to a business problem in COBOL language, and then the processing system generates machine-language instructions capable of performing the operations described by the programmer's statements.

COBOL LANGUAGE

The English-language statements of COBOL provide a relatively machine-independent method of expressing a business-oriented problem to a computer. Commonly used nouns, verbs, and connectives are used in the procedural portion of a COBOL program to construct easily understood sentences. The excellent documentation provided by COBOL — problem definition as well as method of solution — enables more than one programmer to work on a particular problem with minimal duplication of effort. (The accompanying illustration depicts a Honeywell COBOL source program.)

To complement the modularity of the Series 200 hardware, the various COBOL versions implement a set of language modules, expanding the features of COBOL as the machine capacity is increased. This design approach allows the COBOL user to enhance the power of the source language and to produce larger object programs as the need arises.



COBOL PROGRAMMING FORM

```
IDENTIFICATION DIVISION.
PROGRAM-ID. UPDATE.
DATE-WRITTEN. FEBRUARY 10, 1965.
DATE-COMPILED. FEBRUARY 11, 1965.
AUTHOR. JOHN PARNELL.
INSTALLATION. JONES DEPARTMENT STORE.
SECURITY. CONFIDENTIAL TO COMPANY.
REMARKS. THIS PROGRAM READS PURCHASE RECORDS FROM A MAGNETIC
    TAPE FILE, UPDATES AND EDITS INFORMATION IN EACH RECORD, AND
    PRODUCES A TOTAL-RECORDS MAGNETIC TAPE FILE; ERROR PRINTOUTS
    ARE PROVIDED WHERE NECESSARY.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. MODEL-200
OBJECT-COMPUTER. MODEL-200; MEMORY SIZE 16384 CHARACTERS.
    6 TAPE-UNITS, 1 READER-PUNCH, 1 PRINTER.
SPECIAL-NAMES. CONSOLE-TYPEWRITER IS ERROR-PRINTER.
INPUT-OUTPUT SECTION.
FILE-CONTROL. SELECT PURCHASE-FILE, ASSIGN TO TAPE-UNIT AA, TAPE-UNIT
    AB. SELECT TOTAL-FILE, ASSIGN TO TAPE-UNIT AC, TAPE-UNIT AD.
I-O-CONTROL. APPLY TRAILING-COUNT ON PURCHASE-FILE, TOTAL-
    FILE.

DATA DIVISION.
FILE SECTION.
FD  PURCHASE-FILE; BLOCK CONTAINS 50 RECORDS; LABEL RECORDS
    ARE STANDARD; VALUE OF IDENTIFICATION IS "PURCHASE"; DATA
    RECORD IS PURCHASE-RECORD.
01  PURCHASE-RECORD.
    02  LOCATION NUMBER; PICTURE IS 99999.
    02  ORDER NUMBER; PICTURE IS 9999.
    02  STOCK-NUMBER; PICTURE IS 99999V99.
    02  UNITS-ON-HAND; PICTURE IS 999.
    02  UNIT-PRICE; PICTURE IS V99.
    02  UNIT-CODE PICTURE IS XX.
WORKING-STORAGE SECTION.
77  ERROR-INDICATOR-1 PICTURE IS S9, VALUE IS ZERO.
01  LABEL-RECORDS.
    02  DATE, OCCURS 5 TIMES.
    03  MONTH PICTURE IS 99.
    03  YEAR PICTURE IS 99.

PROCEDURE DIVISION.
BEGIN SECTION.
OPEN-FILES. OPEN INPUT PURCHASE-FILE; OUTPUT TOTAL-FILE.
COMPUTING SECTION.
READ-LOOP. READ PURCHASE-FILE; AT END GO TO END-PROCESSING.
    IF STOCK-NUMBER IS NOT NUMERIC GO TO BAD-STOCK-NUMBER.
COMPUTE-TOTAL-PRICE. MULTIPLY UNITS-ON-HAND BY UNIT-PRICE
    GIVING TOTAL-PRICE; ON SIZE ERROR GO TO EXCEEDS-ROUTINE.
    MOVE TOTAL-PRICE TO EDITED-PRICE.
TRANSFER-DATA. MOVE PURCHASE-RECORD TO TOTAL-RECORD.
ERROR-SWITCH. GO TO PRODUCE-OUTPUT.
PRODUCE-OUTPUT. WRITE TOTAL-RECORD. GO TO READ-LOOP.
ERROR-1 SECTION.
BAD-STOCK-NUMBER. ALTER ERROR-SWITCH TO PROCEED TO
    MARK-STOCK-NUMBER. GO TO COMPUTE-TOTAL-PRICE.
```

The IDENTIFICATION DIVISION identifies the source program and provides optional documentation information.

The ENVIRONMENT DIVISION specifies the processor on which the source program is to be compiled, the configuration on which the object program is to be executed, and the relationships between data files and input/output media. The CONFIGURATION SECTION contains three paragraphs which deal with the over-all specifications of the processors involved and equate actual hardware names with mnemonic names supplied by the programmer. The INPUT/OUTPUT SECTION consists of two paragraphs which identify each file and specify input/output techniques, respectively.

The DATA DIVISION describes the data to be processed by the object program. It contains a FILE SECTION which describes the files used. There may be a WORKING STORAGE SECTION which allocates memory space for the storage of intermediate results.

The PROCEDURE DIVISION describes the procedures to be used in processing the data described in the DATA DIVISION; it contains all the necessary steps to solve a given problem. Procedures are written as sentences which are combined to form named paragraphs. Likewise, paragraphs may be combined to form sections. Paragraph and section names are assigned by the programmer so that control may be transferred from one section or paragraph to another.

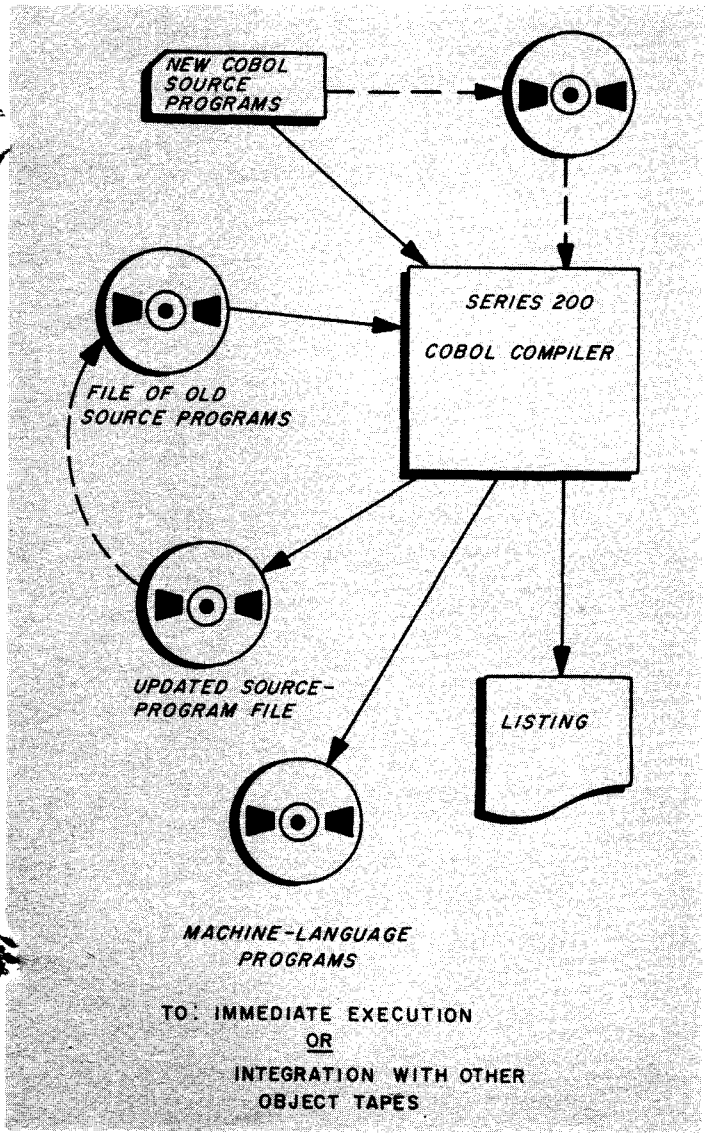
THE COBOL COMPILERS

The Series 200 COBOL compilers are syntax-directed. The smallest version operates in a system consisting of four magnetic tape units, a card reader, an on-line printer, and a 16,384-character memory. (Most competitive compilers possessing a comparable set of language elements require a memory about three times this size.) Honeywell COBOL compilers are known for their high performance, and the Series 200 compilers are no exception: Compile times for typical programs are on the order of one to two minutes with the smallest version.

In addition to providing an expanded language capability, the larger compiler versions enable the processing of larger source programs. COBOL compilers are available for memory sizes of 32,768 characters and larger.

The Series 200 COBOL compilers possess several significant operating features:

- *Maintenance facilities for source-language files.*
- *Fast diagnostic scan.*
- *Job-oriented operating system.*
- *Dynamic reassignment of read/write channels.*
- *A variety of checkout aids such as a data distribution system, dynamic and static dumping facilities, etc.*



SERIES 200 FORTRAN

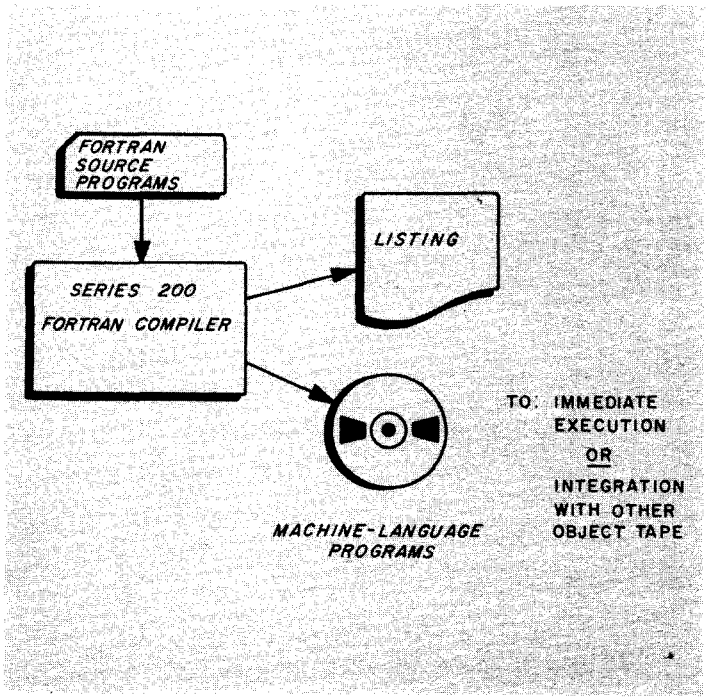
Fortran consists of two basic elements: a source language (Fortran IV) whose structure closely resembles the language of mathematics, and a compiler which translates the statements and formulas written in the source language into a machine-language program. Series 200 Fortran is an upward-compatible set of compiler versions that provides increasingly powerful language elements and operating characteristics for the larger equipment configurations.

FORTRAN LANGUAGE

Programs are written directly as algebraic expressions and arithmetic statements. Additional statements, such as transfer, decision, indexing, and input/output statements, control the processing of the algebraic expressions. The smallest compiler version translates a major portion of Fortran IV, including logical statements and testing, data initialization, labelled COMMON areas, and type statement declarations. Even more sophisticated language elements are accepted by the larger versions.

THE FORTRAN COMPILERS

All Series 200 Fortran compilers are designed for rapid compilation and optimum efficiency of object coding. Translated programs can be combined with other previously compiled and assembled programs and immediately executed to obtain fast results. The smallest version requires as few as 16,384 characters of memory, plus four magnetic tape units, a card reader, card punch, and printer. Larger versions, which exploit the added features and instructions of the scientific hardware option, can process programs utilizing very large core storage capacities, up to 524,000 characters. Special features of the Fortran compilers include object code optimization and a highly sophisticated diagnostic system.



GENERALIZED DATA MANIPULATION PROGRAMS

In addition to program preparation aids, Honeywell also provides an extensive array of software to relieve the user of the tedious and complex task of programming such common jobs as sorting, input/output operations, and report generation. Many of these generalized programs are offered in two or more versions, each specifically tailored to take fullest advantage of a particular range of equipment configurations.

MAGNETIC TAPE INPUT/OUTPUT PACKAGE — TIPTOP

Implemented to handle both Honeywell and competitive data conventions, the tape input/output control package provides object code, as directed by macro instructions, to perform the following functions: reading and writing tape records, blocking and unblocking of items within records, opening and closing files, and detection and automatic correction of errors (see accompanying illustration). Both fixed-length and variable-length records are handled by TIPTOP.

PAPER TAPE INPUT PACKAGE — TOPPER

The input package for paper tape systems — TOPPER — can handle 5-, 6-, 7-, or 8-level paper tapes. TOPPER performs all input functions stated above for TIPTOP with the added capability for data editing. Exits are also provided to a user-supplied code-conversion table.

UTILITY ROUTINES

Honeywell provides a growing library of utility routines to extend the capabilities of the standard operating aids provided for Series 200 users. Descriptions of some of these routines follow.

THE UPDATE AND SELECT PROGRAM

This program accepts as input a transaction program tape, an old program master file, and a deck of director and correction cards. The transaction and master file tapes can contain machine-language programs produced by the Easycoder Assembly System and the COBOL and Fortran compilers, as well as Honeywell software programs. The Update and Select Program can perform the following functions under control of parameters specified on the input director cards:

PROGRAM MASTER FILE UPDATE — Programs from the old master file are deleted, corrected, or left unchanged, and new programs are added from the transaction tape. Updating can include changing of program job assignments. A new program master file tape is produced.

PROGRAM SELECTION — A run tape which contains selected programs from the new program master file is produced for use in checkout or production operations.

DIRECTORY LISTING — A directory is printed which includes a separate listing of programs in the new program master file and on the production run tape.

DYNAMIC TAPE AND MEMORY DUMP ROUTINES

These routines, particularly valuable when debugging programs, provide automatic, "on-the-fly" recording of the contents of memory and of magnetic tape files. Calls to these routines may be programmed in advance by use of macro instructions or initiated at object time by the operator.

PATCH ROUTINE

The use of Patch enables octal changes (or corrections) to be made to specified programs at object program execution time. The changes occur in core memory only; they do not affect the object program stored on the run tape.

THOR — TAPE HANDLING OPTION ROUTINE

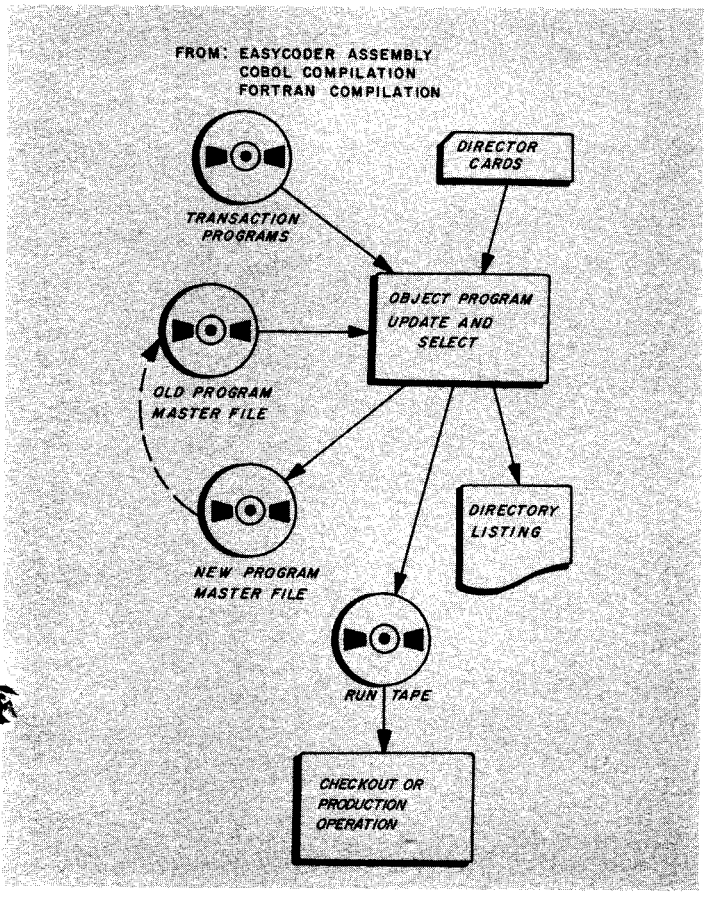
THOR is a set of general tape-handling and correction routines for use with the Series 200. Under the direction of parameters supplied by the operator from punched cards, paper tape, or the control panel, THOR can perform nine separate functions. These include:

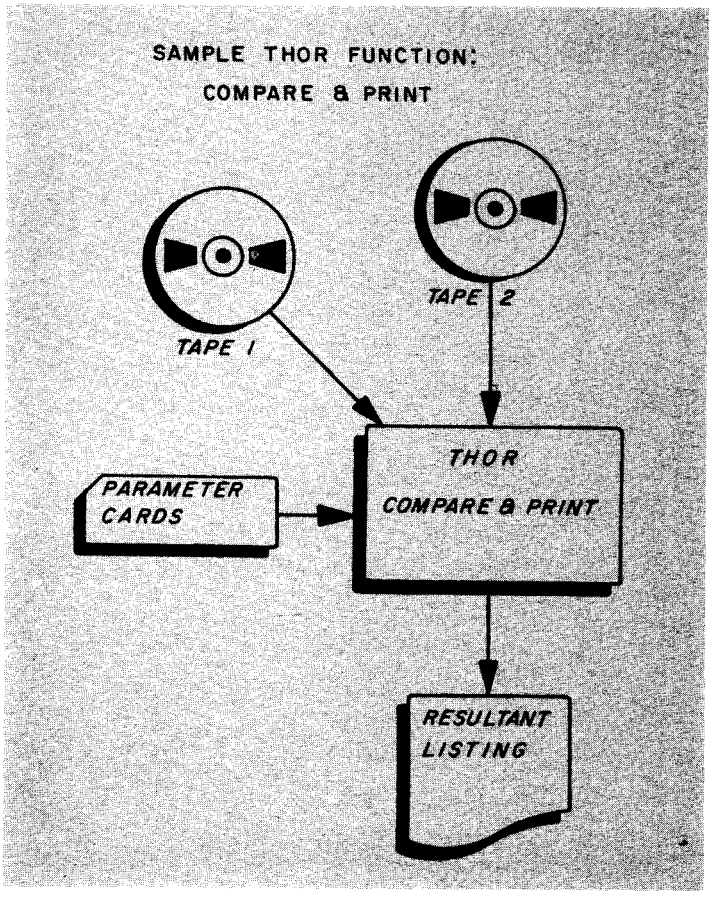
COMPARE AND PRINT — A specified number of records from each of two tapes are compared, record for record, with all non-identical records printed in either alphanumeric or octal mode.

LOCATE — A tape is searched for the first occurrence of specified information.

CORRECT AND COPY — A designated record is copied from one tape to another with specified corrections.

In manipulating magnetic tapes, either a record-counting method or a file identification method may be employed. The file option provides added convenience in that it permits operation over an entire tape or file, rather than over a specified number of records.



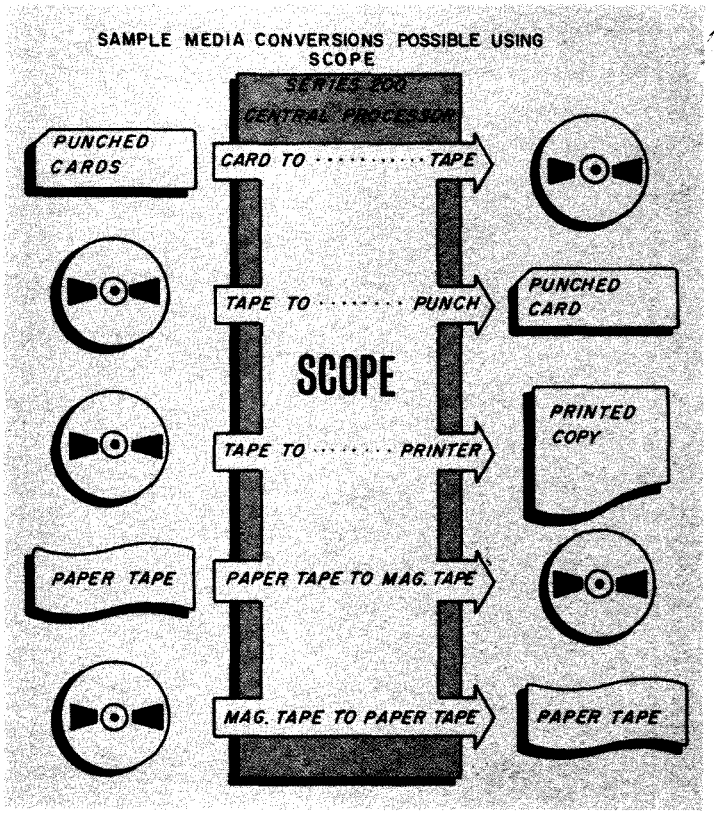


SCOPE — SYSTEM FOR COORDINATION OF PERIPHERAL EQUIPMENT

SCOPE consists of a group of independent coroutines which control the automatic transfer of data between pairs of peripheral devices such as magnetic tape drives, punched card equipment, paper tape equipment, and printers.

A variety of data conversion operations can be performed by SCOPE (see the accompanying illustration). The degree of simultaneity achieved in performing combinations of these conversion operations depends upon the Series 200 processor employed.

"Own-coding" routines (prepared in Easycoder assembly language) may be included in a SCOPE deck to perform such functions as editing and unblocking of records. For example, if it is desired to edit records coming from or going to various terminal devices, own-coding can be inserted into each of the terminal device coroutines at specified points.



SUMMARY

Honeywell offers a full complement of programming and operating aids designed to assist the user in taking fullest advantage of superior Series 200 Hardware. This software has the following qualities:

COMPREHENSIVENESS

Included are all the functions required in normal data processing: assembly system, business- and science-oriented compilers, conversion systems, tape and random access sorts and I/O programs, mathematical subroutines, report generator, object program maintenance and operating system, and an extensive array of utility programs.

FLEXIBILITY

Modular design ensures that programs and systems are available for all Series 200 sizes and configurations.

POWER

Performance is maximized by combining source languages encompassing a wide range of functions with highly efficient processing techniques.

SIMPLICITY

Easy-to-use source languages and simple operating procedures enable programmers and operators to take maximum advantage of the throughput and efficiency built into the hardware.

INTERCHANGEABILITY

Object programs produced by both compilers and by the assembly system, as well as Honeywell software programs, can be intermixed and processed by a common operating system.

INTRA-SERIES COMPATIBILITY

Software and software-produced object programs which run in the smallest Series 200 processor can run to advantage in the largest system configuration.

JOB ASSIGNABILITY

Programs can be assigned during assembly or compilation to specific jobs and called at checkout or production time either singly or, for automatic serial performance, by job.

FULL BACK-UP AND SUPPORT

The well staffed education, systems service, and documentation facilities of Honeywell EDP all contribute to a continuing back-up program for all Series 200 programming and operating aids.

Honeywell

ELECTRONIC DATA PROCESSING

SALES OFFICES AND DATA CENTERS
IN PRINCIPAL CITIES OF THE WORLD

DSA-130
25265
10265

Printed in U.S.A.
WP-8050